REMARKS

The foregoing Amendment After Final and the following Remarks are submitted in response to the Final Office Action issued on July 7, 2006 in connection with the above-identified patent application.

Interview Summary

Applicant appreciates the courtesies extended by Examiner Shiferaw during a telephonic interview with Applicant's undersigned representative on October 25, 2006. During the interview, Applicant's undersigned representative summarized the invention and the differences between the claimed invention and the cited prior art patent to Vu et al. The Examiner invited Applicant to submit further amendments and comments for consideration that would support the comments made by Applicant's undersigned representative. The present Amendment response is being submitted for this purpose.

Claim Amendments

Independent claims 15, 25, 31, and 41 have been amended to overcome the rejections of these claims and dependent claims 16 and 32 have been canceled. Claims 26 and 42 also have been amended to correct typographical errors. Upon entry of this Amendment Response, claims 15, 20-27, 31, and 36-43 will remain pending in the present application. Independent claims 15, 25, 31, and 41 have been amended to clarify the transitioning from the normal mode to the preferred mode and vice-versa for authentication/verification of the instantiated application. The amended claims are thus believed to better support the arguments previously submitted with respect to the Vu et al. patent, many of which are repeated herein. Applicants respectfully submit that no new matter has been added to the application by the proposed amendments.

Prior Art Rejection

As noted in previous amendment responses, the present invention is directed toward the problem that for a computing device such as a portable computing device to be trusted in the context of a rights management architecture, the portable device and the processor thereon must be of a type that substantially completely prevents a content thief from performing nefarious acts that would allow obtaining of content therein in an unencrypted form or allow obtaining decryption keys. Thus, according to the present

invention, the processor is a secure processor and is constructed to run only authorized code, and is operated to maintain a strict cryptographic separation between applications that may be instantiated thereon.

The secure processor is operable in a normal mode and a preferred mode, where a security kernel can access a locally accessible CPU key only during the preferred mode. The security kernel employs the accessed CPU key during the preferred mode to authenticate and instantiate a secure application such as a rights management system, a banking / financial system, etc. on the portable device. Importantly, whatever the application may be, such application is not instantiated until authenticated by the security kernel during the preferred mode, and is thereafter instantiated while in the preferred mode, after which normal mode is entered, at which time the instantiated application may be employed.

The accessed CPU key is typically a key that is employed by the security kernel to decrypt one or more encrypted security keys for the application instantiated. The CPU key is accessible only by the security kernel and only during the preferred mode, and is the key to unlocking or decrypting the secrets identified with each application, and therefore must be well-protected.

The Examiner has finally rejected claims 15-16, 20-27, 31-32, and 36-43 under 35 USC § 103 as allegedly being obvious over Vu et al. (U.S. Patent No. 6,557,104) in view of Ginter et al. (U.S. Patent No. 5,892,900) and Mirov et al. (U.S. Patent No. 6,138,236). Applicants respectfully traverse the § 103 rejection insofar as it may be applied to the claims as amended.

Independent claim 15 of the present application as amended recites a method for a secure processor to instantiate and authenticate a secure application thereon by way of a security kernel. In the method, the secure processor is powered on into a normal mode, and receives an instruction to instantiate the application after being powered on and while being in the normal mode. After receiving the instruction to instantiate the application, the secure processor transitions from the normal mode to the preferred mode upon a non-power-up executed CPU reset, where a security key of the processor is accessible while in the preferred mode and instantiates and runs a security kernel while in such preferred mode. Thereafter, the security kernel while in the preferred mode accesses the security key and applies same to decrypt at least one encrypted key for the application, stores the decrypted key(s) in a

location where the application will expect the key(s) to be found, and authenticates the application on the processor. Significantly, the application is instantiated while in the preferred mode and only after the security kernel has authenticated such application.

The secure processor then transitions from the preferred mode to the normal mode after the security kernel authenticates the application and the application has been instantiated, where the security key is not accessible while in the normal mode. The application as instantiated during the preferred mode is thus available for use during the transitioned-to normal mode. Thus, the security kernel allows the processor to be trusted to keep hidden the key(s) of the application. The security kernel employs the accessed security key during the preferred mode to authenticate / verify the application prior to instantiation thereof.

Independent claim 31 recites the subject matter of claim 15, albeit in the form of a computer-readable medium.

As amended, then, and to summarize, in the present invention as recited in claims 15 and 31, the initial transition from the normal mode to the preferred mode occurs some time after power-up based on a non-power-up, executed CPU reset, as opposed to a CPU reset that might automatically occur during power-up. In addition, the application is not instantiated until the processor is in the preferred mode and authentication takes place, and the application as instantiated remains even after the transition from preferred mode to normal mode.

As was previously pointed out, the Vu reference discloses a secure processor where, during run-time, an already-instantiated application requiring access to a secure service invokes a security routine that in turn invokes a security mode by way of a high-level security interrupt which cannot be otherwise invoked. As best set forth at column 5, lines 24-41, once in the security mode (which the Examiner appears to equate with the preferred mode of the claims), a security function is invoked to access secrets and data from an otherwise inaccessible storage location, and the security function performs appropriate security functionality based on such secrets and data, such as for example encryption and decryption, password validation, user authentication, etc.

As the Examiner notes in a Response to Arguments at page 2 of the Final Office Action, the Vu reference may be interpreted to disclose transitioning between modes

during a CPU power-up. Accordingly, in amending claims 15 and 31 of the present application, Applicants have specifically recited that the CPU reset is a non-power-up executed CPU reset, such as that which may be programmatically triggered. Thus, Applicants respectfully submit and the Examiner has in effect conceded that the Vu reference does not disclose such a CPU reset.

That said, in the Vu reference, the Vu application is already running and initiates the security function when services of a secure entity are required (see, for example, step 20 of Fig. 2). In contrast, the present invention as recited in the claims as amended does not even allow the application to be instantiated until the application has been authenticated by the security kernel during the preferred mode, where such preferred mode is not transitioned to until after a non-power-up CPU reset, and where the instruction to instantiate the application occurs after power-up and while in the normal mode, i.e., before the CPU reset. Thus, the Vu reference does not disclose or even suggest an application that is instantiated only after being authenticated in the manner recited in claims 15 and 31.

The Examiner cites to the Ginter reference as teaching a cache or the like from which data is erased when transitioning between modes such that sensitive data employed during one mode is not available during a following mode. Also, the Examiner now cites to the Mirov reference as teaching instantiating an application after authenticating same. All this notwithstanding, the Ginter and Mirov references like the Vu reference also fails to disclose or even suggest an application that is instantiated only after the application has been authenticated by the security kernel during a preferred mode, where such preferred mode is not transitioned to until after a non-power-up CPU reset, and where the instruction to instantiate the application occurs after power-up and while in the normal mode, i.e., before the CPU reset, all as recited in claims 15 and 31. Moreover, neither the Vu reference nor the Ginter reference nor the Mirov reference teaches or even suggests transitioning between modes by way of a non-power-up executed CPU reset, as is recited in claims 15 and 31. In Vu in particular, and as was previously pointed out, the transition is instead accomplished by way of an interrupt. Accordingly, even if the teachings of these references could have been combined by one skilled in the art as the Examiner suggests, the claims method and computer readable medium would not have resulted.

Independent claim 25 recites a method for a secure processor to instantiate one of a plurality of available secure applications thereon by way of a security kernel. In the method, a chooser value is set to a value corresponding to a chooser application upon power-up, and a preferred mode is entered upon a power-up CPU reset and instantiating the security kernel. The security kernel determines that the chooser value corresponds to the chooser application and therefore authenticates such chooser application, after which such chooser application is instantiated.

After the chooser application is instantiated, a normal mode is entered and the chooser application presents the plurality of available applications for selection by a user. Upon receiving a selection of one of the presented applications to be instantiated, the chooser value is set to a value corresponding to the selected application. Thereafter, a CPU reset is executed and the preferred mode is re-entered, and the security kernel is instantiated. The security kernel then determines that the chooser value corresponds to the selected application and therefore authenticates the selected application, after which such selected application is instantiated. Normal mode is then re-entered after the selected application is instantiated and run. Thus, the security kernel allows the processor to be trusted to keep hidden a secret of the chooser application and a secret of the selected application. As amended, claim 25 emphasizes among other things that two CPU resets take place, including a first, power-up reset and a second, non-power-up executed reset.

Independent claim 41 recites the subject matter of claim 25, albeit in the form of a computer-readable medium.

Applicants reassert the arguments with respect to claims 15 and 31 to claims 25 and 41. Notably, inasmuch as the Examiner has already in effect conceded that the Vu reference only discloses a single CPU reset at power-up, Applicants respectfully submit and the Examiner must concede that the Vu reference does not disclose the use of the two CPU resets of claims 25 and 41, one with regard to a chooser application and another with regard to a selected application as chosen by way of the chooser application.

In addition, Applicants again point out that the Vu reference does not disclose or even suggest the use of a chooser application to choose a selected application, let alone the use of a secure kernel that operates in the detailed manner set forth in claims 25 and 41. Instead, in the Vu system the application itself initiates authentication, and therefore cannot

be instantiated only after such authentication. Moreover, Vu does so based on only a single CPU reset. In addition, the Ginter reference and the Mirov reference like the Vu reference also fail to disclose or even suggest the combined use of such a chooser application, selected application, and pair of CPU resets in the manner recited in claims 25 and 41. Moreover, neither the Vu reference nor the Ginter reference nor the Mirov reference teaches or even suggests transitioning between modes in the particular manner recited in claims 25 and 41, i.e., from normal mode to preferred mode to normal mode to preferred mode to normal mode.

In addition, Applicants once again respectfully submit that the Vu reference dopes not suggest employing a chooser application, a chooser value, and the use thereof to securely choose and instantiate one of a plurality of applications on a secure processor in the manner recited in claims 25 and 41. In particular, the Vu reference does not at all disclose or even suggest switching between modes as recited to first load and then operate a chooser application, employ same to select a chooser value corresponding to a chosen application, and then load and operate a chosen application in the manner recited in claims 25 and 41.

Accordingly, Applicants respectfully submit that the Vu, Ginter, and Mirov references cannot be combined to make obvious the invention recited in claims 15, 25, 31, and 41 or any claims depending therefrom. Accordingly, and for all the aforementioned reasons, Applicants respectfully request reconsideration and withdrawal of the § 103 rejection.

Conclusion

In view of the foregoing discussion, Applicants respectfully submit that the present application, including claims 15, 20-27, 31, and 36-43, is in condition for allowance, and such action is respectfully requested.

Respectfully submitted,

Date: November 7, 2006

/Michael P. Dunnam/
Michael P. Dunnam
Registration No. 32,611

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439